Cross-Layer Mischief in Networked Control Systems

Gregory Nazario, Michael Rosen, Lawrence Jackson, Michael Hankowsky
Carnegie Mellon University
Electrical & Computer Engineering
Pittsburgh, USA
{gnazario, mrrosen, lmjackso, mhankows}@andrew.cmu.edu

***Abstract*—SCADA (supervisory control and data acquisition) systems are becoming increasingly prevalent, especially wirelessly networked control systems. [citation] The integration of wireless networking into the operation of these SCADA systems is good for many reasons, but these integrations often ignore security as a priority. [citation] Our goal is to better understand the capabilities and the vulnerabilities these systems characteristically possess. In order to do this we will explore the use of cross-layer attacks on networked control systems and develop a framework for testing these attacks. We have developed testbed consists of a simulated network, a simulated factory driven by a matlab-ran Tennessee-Eastman process and the interface between these systems. Our research has confirmed that these vulnerabilities are exploitable and we have established this in our testbed. These vulnerabilities point to the necessity for a change in the way we look at securing SCADA integrated wireless networks.**

I. Introduction

The growing prevalence of SCADA systems have brought much needed attention to the class of cyberphysical threats. This paper examines the specific effects and testing of cyberphysical threats in wireless networking. Our intent is to simulate a networked control system similar to that of what one might find in a chemical plant. We approached Stuxnet destroyed a fifth of Iran's nuclear centrifuges. This is one simple example of SCADA SCADA Systems ... [more on scada]

A. The Framework

Subsection text here.

1) Subsubsection Heading Here: Subsubsection text here. at

II. Related Work

III. Approach

Our goals were to create a modular framework that:

1) allows for extension by other developers, but without specific domain knowledge
2) provides for consistent parameterized simulation
3) allows for diverse simulations tailored to realistic SCADA conditions

A. Simulated Network

The network of our simulation is designed to give a real-time nature to the data transfer between the Tennessee Eastman(TE) physical simulation and the TE Controller. Both the controller and simulation are written in Matlab and connected to our network with the OMNeTBridge Class. (These are described in later sections.)

The network is based primarily on two factory level components, sensors and actuators, that send data to a remote facility that sends back control signals. This is based on most SCADA systems that need remote facilities to monitor their data. These signals are sent over a simulated "Internet" that allows accurate packet loss, bit errors and data rates.

Due to limitations of the default Inet packets we are not actually sending the data through the network. Rather, the packets the controller receives act as a trigger to have the controller update the data from that sensor, and send an update to the actuators if need be.

B. Matlab Simulation

We chose to use MATLAB simulation for the physical system and it's controls, as there are many chemical process models currently implemented in MATLAB. This allows for the ease of swapping out current MATLAB models for new ones.

1) Tennessee Eastman: In order to keep the complexity down, we decided to model the chemical process in the plant as a simplified Tennessee Eastman problem. Due to this approach, we looked to previous work in the field, and found many models already modeled in a hybrid of FORTRAN and MATLAB. However, this introduced more complexity, due to using three different programming languages to implement what should be simpler. Therefore, we would build a custom MATLAB class that contained the same functionality. This includes a both the simulation of the actual Tennessee Eastman system, and the steady state controller calculations.

C. C++ to Matlab Bridge

IV. Implementation

A. Simulated Network

The network of our simulation is designed to give a real-time nature to the data transfer between the Tennessee Eastman(TE) physical simulation and the TE Controller. Both the controller and simulation are written in Matlab and connected to our network with the OMNeTBridge Class.

B. Controller

The controller is essentially a server module that

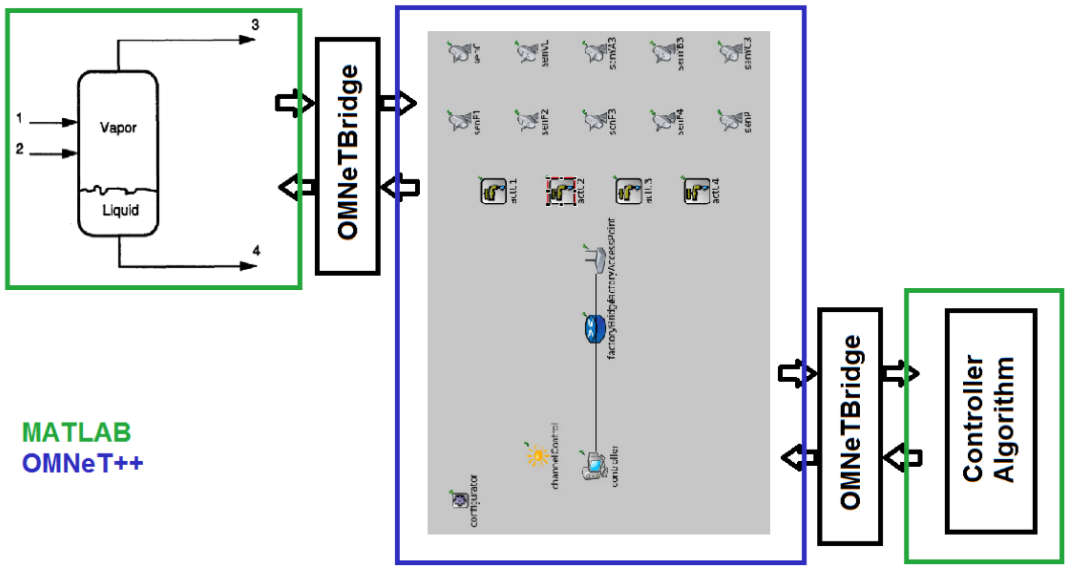

Fig. 1. System Diagram.

1) Actuator: The actuator acts as a server which receives an update packet from the controller, grabs its appropriate data from the controller OMNeTBridge and then passes it to the correct function. When

2) Sensor: The Sensor is a relatively simple module that sends an update signal based on a timer. This is to model the refresh rate of most sensors. We have assumed that all sensors and network controllable and are able to tirelessly connect to a local access point via 802.11 Wireless LAN.

These modules send a TCP packet to a known IP address which in turn triggers the controller to update. This is a non-ideal method of updating as it does not allow correct implementation of attacking the data on the network. Ideally INET's TCP packets would allow us to send data to them, however as some modules are currently implemented deep copies are not done of messages or inappropriate casts are used. Future work should be planned to modify these modules or find a more correct implementation of INET messages that allow custom data fields to be created and sent.

C. MATLAB Simulation

Our MATLAB both simulated a system and a controler of the Tennessee Eastman problem, and used a custom packet interface between C++ and MATLAB as explained below.

1) Tennessee Eastman: We built a custom simplified Tennessee Eastman model in MATLAB. This was based off of a preexisting FORTRAN model, but was re-implemented in MATLAB for ease of use and reduced complexity . Our model contains variable vectors, an input vector, a state vector, and an output vector. Then by using the equations provided in Ricker, the simulation simply takes in the input vector from OMNeT++, and outputs the output vector, which includes the sensor data, to OMNeT++ via the MATLAB bridge.

On the other side of the network, we also implemented a controller in MATLAB for the Tennessee Eastman system. This used steady state calculations to determine the optimal controls. The controller would input the output vector from the Tennessee Eastman system via the MATLAB bridge, and output a set of input vectors to be transmitted across the network into the MATLAB bridge.

D. C++ to MATLAB Bridge

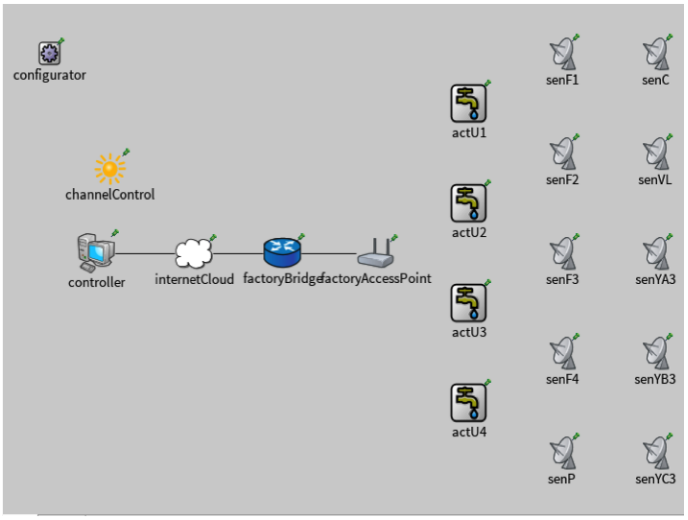

Fig. 2. Network Diagram.

V. Challenges

A. Omnet++

B. Matlab

C. Bridge

VI. Evaluation

A. Metrics

We will consider three metrics when evaluating the results of our experimentation: sensor data, control signals and packet throughput.

1) Sensor Data: We consider Sensor Data because this includes crucial data about what is actually occurring in the simulated plant process (Tennessee-Eastman, in this case).

2) Control Signals: We consider the Control signals because these signals are meant to be directing the process, and how they react when parameters of the simulation are changed a crucial to our understanding of how various attacks effect our simulation.

3) Throughput: We use the packet throughput as an evaluation metric because this data is extremely relevant in denial of service type attack we have implemented.

VII. Results

A. Baseline

B. DDoS

C. Spoofing

VIII. Discussion

The project was relatively successful with the completion of the simulation framework and the implementation of some basic attacks. While we would have liked to include more attacks and improved the framework to more accurately model modern SCADA networks, the construction of the communication bridges and basic example of a network was a significant project and can be expanded to larger models and simulations if needed. The pipe and bridge system can used used for arbitrary MATLAB or OMNeT++ models, thus our example of a distributed SCADA network controlling a Tennessee-Eastman process is just one example. The framework was constructed to be dynamic, thus allowing for plug and play or various network and MATLAB models.

In the course of development, several challenges impeded work on the framework. Working through C++'s abandonment of the standard cast and void* generic type from C was somewhat frustrating, meaning converting floats became difficult, as well as dealing with development for both 32 and 64-bit machines. Aside from these minor technical issues, some of the biggest problems arose from the way in which INET and OMNeT++ model data transfer. As mentioned about, OMNeT++ and INET only attempt to create a timing-accurate model for various network protocols and hardware. Data-transfer is relatively absent and thus, when sending packets across the network model containing data from either the sensors or controller, the actual values being sent where often lost. Thus, ensuring data was able to get through the network was a significant challenge as the INET framework is very large and finding the few lines which destroy and recreate the packet was difficult. While we did get the simulation running, INET should be capable of preserving objects across the network, and it is a mystery why it does not.

While we did not model an exact SCADA system with industry software, the dangers illustrated in our results are still of significant concern. Throwing the process into an unstable state was not difficult and control system in general require real-time control. Delaying or spoofing signals in such systems can lead to instability. In the case of nuclear power or other chemical processes, these instabilities can be disastrous. As more and more control systems find their way onto the Internet, remote attacks become easier and physical access to the plant or system becomes unnecessary. Attackers from across the country or even other nations can target and disrupt control systems for critical infrastructure. Thus, a significant threat exists and more work needs to be done to ensure the safety and security of the various critical systems now on the Internet.

IX. Future Work

A large goal of this project was to develop a framework that would enable further research to be done in this space. What we've provided makes it very easy to develop different network configurations for corresponding process simulations in Matlab. There are three major fields of further work that we propose: extending the framework, experimenting with more attacks and validating configurations.

A. Extending the Framework

The framework is functionally complete, but there are additions that could be made. More modules could be adapted to support a greater range of network simulations. The addition of primitives to support cryptography, or at the least simulated cryptography, would enhance the realism of this framework and allow for truer to life attacks.

B. Experimentation

A large body of future work could easily consist of just experimentation with different types of misbehavior and exploitation.

C. Validation and Research

It would be valuable to do further research into the common configurations of networked control systems in industry. This information would certainly help focus future experimentation and development.

X. Conclusion

We have developed a

Acknowledgment

The authors would like to thank Bruce DeBruhl for guiding our research and for helping us to refine our ideas.

REFERENCES

[1] Siaterlis et al., *EPIC: A Testbed for Scientifically Rigorous Cyber-Physical Security Experimentation*, Vol 1, No.2 IEEE Transactions on Emerging Topics in Computing, 2013.

[2] Mo et al., *Cyber-Physical Security of a Smart Grid Infrastructure*, Vol. 100, No.1 Proceedings of the IEEE, 2012.

[3] Cardenas et al., *Challenges for Securing Cyber Physical Systems*, 2009.

[4] N. L. Ricker, *Model predictive control of a continuous, nonlinear, two-phase reactor*, Seattle, WA: University of Washington, 1993.

[5] A. Hayes, *Network Service Authentication Timing Attacks*, IEEE Computer and Reliability Societies, 2013.

[6] H. Beitollahi, G. Deconinck, *A Dependable architecture to mitigate distributed denial of service attacks on network-based control systems*, Vol.4, International Journal of Critical Infrastructure Protection, 2011.

[7] S. Radosavac, N Benahammar and JS Baras, *Cross-layer attacks in wireless ad hoc networks*,Princeton, New Jersey: Conference on Information Sciences and Systems, 2004.

[8] S. Amin, A. Cardenas and S.S. Sastry, *Safe and Secure Networked Control Systems under Denial-of-Service Attacks*, Berlin, Germany: Springer-Verlag, 2009.

[9] Hashimoto et al., *Safety securing approach against cyber-attacks for process control system*, Vol. 57,Nagoya, Japan: Computers and Chemical Engineering, 2013.

[10] Y. Mo and B. Sinopoli, *Secure Control Against Replay Attacks*, 47th Illinois, USA: Annual Alerton Conference, 2009.

[11] S. McLaughlin, *Securing Control Systems from the Inside: A Case for Mediating Physical Behaviors*,IEEE Security & Privacy, 2013.

[12] X. Chen, K. Makki, K. Yen and N. Pissinou, *Sensor Network Security: A Survey*, Vol. 11, No. 2IEEE Communications Surveys & Tutorials, 2009.